FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO


# EXERCÍCIOS DE PROGRAMAÇÃO EM LÓGICA COM RESTRIÇÕES

LUÍS PAULO REIS

LICENCIATURA EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO

PROGRAMAÇÃO EM LÓGICA - 3º ANO

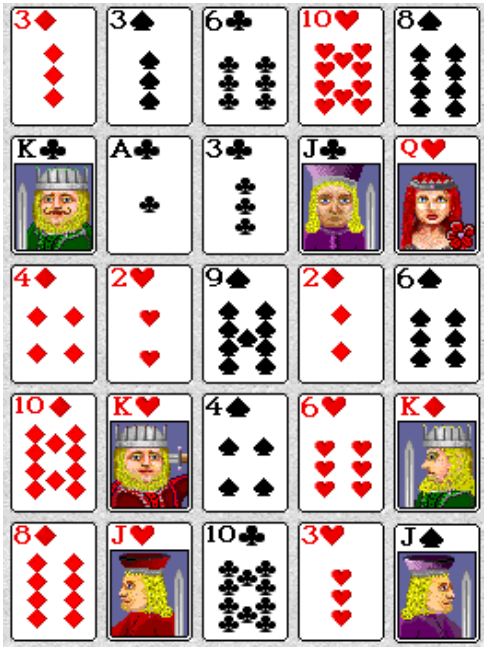
NOVEMBRO DE 2004

# Programação em Lógica Com Restrições – Problemas de Optimização

## PLOP 1: O Poker do Nabais

O Nabais adora jogar Poker. Quando não tem parceiros disponíveis para jogar, ele diverte-se a jogar o jogo como solitário. Sorteia 25 cartas e tenta dispô-las de forma a conseguir as melhores mãos de Poker possíveis na Horizontal e Vertical! A sua tabela de pontuações é a seguinte:

- Um par – 10 pontos
- Dois pares – 20 pontos
- Trio – 40 pontos
- Sequência – 60 pontos
- Cor – 90 pontos
- Full House (trio+par): 150 pontos
- Poker (quatro cartas iguais): 200 pontos
- Straight Flush (sequência de cor): 300 pontos
- Royal Flush (sequência de cor de Ás a 10): 400 pontos



Supondo que o Nabais obteve a seguinte mão:

Consegue fazer um programa em CLP capaz de determinar o melhor arranjo possível para este problema (o Nabais conseguiu manualmente 1050 pontos!).

## PLOP 2: O Carteiro Preguiçoso

Um carteiro com pouca correspondência para entregar e muito tempo para gastar, pois cada vez mais as pessoas utilizam o Email, estabeleceu o objectivo de terminar a sua ronda gastando o máximo de tempo possível enquanto entregava a correspondência na sua última rua. Esta rua é rectilínea, contendo 10 casas equidistantes a uma distância de 10 metros umas das outras. O carteiro anda a uma velocidade mínima de 10 metros por minuto (que é a velocidade que vai utilizar ao longo de todo o seu percurso). Começando pela casa nº1, entregou a correspondência, depois foi até à casa nº10, depois à casa nº2, depois à nº9 e por aí em diante até terminar na casa nº6 onde lhe ofereciam sempre café e bolo. Caminhou no total: 9+8+7+6+5+4+3+2+1 = 45 minutos.

Um dia, contudo, o muito trabalhador carteiro lembrou-se que podia fazer ainda melhor (isto é pior) do que este percurso e elaborou um percurso ainda mais longo que mesmo assim terminava na casa nº6. Qual foi este percurso?

**PLOP 3: Planeamento na Política**

O político Jorge Bosque procura reunir alguns votos para o seu partido através de estratégias menos aconselháveis. Existem três métodos que podem ser utilizados neste campo

1) Convencer votantes individualmente. Isto gasta 2 unidades de tempo e causa uma unidade de sofrimento;
2) Convencer grupos de votantes. Isto permite arranjar 3 votantes de uma só vez, gasta 5 unidades de tempo e causa quatro unidades de sofrimento;

No entanto há também um método não oficial:

3) Insultar grupos específicos de votantes, o que embora faça perder 10 votantes, permite também perder 11 unidades de sofrimento, o que o faz sentir optimamente (o sofrimento dos votantes não é considerado neste jogo…). Esta actividade gasta unicamente uma unidade de tempo pelo que pode ser bastante útil para aliviar o sofrimento!

O Jorge Bosque sabe que necessita de 5 a 6 votantes para ganhar as eleições que se realizam daqui a 10 unidades de tempo (semanas). Que planos pode aplicar para atingir os seus objectivos?

**Solução:**

```
task(1,2,1,1).
task(2,5,4,3).
task(3,1,-11,-10).

make_plan(0,[]) :- !.
make_plan(N,[_|Rest]) :-
    N1 is N - 1,
    make_plan(N1,Rest).

evaluate_plan([],Du1,Pa1,Pe1,Du1,Pa1,Pe1).
evaluate_plan([Id|T],Du1,Pa1,Pe1,Du2,Pa2,Pe2) :-
    task(Id,DuD,PaD,PeD),
    Du is Du1 + DuD,
    Pa is Pa1 + PaD,
    Pe is Pe1 + PeD,
    evaluate_plan(T,Du,Pa,Pe,Du2,Pa2,Pe2).

make_cost(_,[],[]).
make_cost(Sum,[Id | L],[S | T]) :-
    task(Id,_,PaD,_),
    S is Sum + PaD,
    make_cost(S,L,T).

plan(N,Plan) :-
    make_plan(N,Plan),
    domain(Plan,1,4),
    evaluate_plan(Plan,0,0,0,Du,Pa,Pe),
    Pe #>= 5, Pe #<= 6, Du #<= 10, Pa #<= 10,
    make_cost(0,Plan,Cost),
    minimize(labeling([],Plan),Cost).
```

```
% Tests: (The first argument is the length of the plan being searched).
% plan(5,P).
% Found a solution with cost 5
% P = [1, 1, 1, 1, 1]     More? (;)
% no (more) solution.
% plan(4,P).
% no (more) solution.
% plan(3,P).
% Found a solution with cost 6
% P = [1, 1, 2]     More? (;)
% Found a solution with cost 6
% P = [1, 2, 1]     More? (;)
% Found a solution with cost 6
% P = [2, 1, 1]     More? (;)
% no (more) solution.
```

**PLOP 4: Encaixe de Peças**

Suponha os seguintes três blocos:

```
1) AA     2) BB     3)  C
   AA        B         CC
```

e o seguinte tabuleiro (5x2):

```
XXXXX
XXXXX
```

a) Construa um programa em CLP que permita definir a forma de encaixar os três blocos (sem rotações) no tabuleiro referido.

b) Generalize para um tabuleiro de dimensões superiores e mais peças

c) Generalize de forma a permitir rotações e espelhamentos das peças

d) Generalize para qualquer tabuleiro (rectangular com dimensões fornecidas pelo utilizador)

e) Generalize para qualquer tipo de peças

f) Introduza uma função de avaliação que permita calcular a melhor solução considerando que quanto mais abaixo e à direita no tabuleiro estiverem as peças 1 e 2 melhor será a solução.

g) Generalize para o caso de blocos e tabuleiros tridimensionais

h) Caso o problema não seja resolúvel, procure a solução com o mínimo de sobreposições

i) Generalize para o caso em que em cada quadrícula do tabuleiro podem ser sobrepostas N peças (sendo N especificado pelo utilizador).

j) Resolva problemas em que o objectivo seja minimizar N, encaixando todas as peças no tabuleiro.

**PLOP 5: Tarefas Domésticas**

Um jovem casal, João e Maria, pretendem dividir as tarefas domésticas de modo que cada um tenha apenas 2 tarefas, minimizando o tempo total gasto nelas. A eficiência deles varia com as tarefas, sendo o tempo gasto por semana por cada nas 4 tarefas previstas apresentado na tabela abaixo.

| | Compras | Cozinha | Lavagem | Limpeza |
|---|---|---|---|---|
| Maria | 4.5 | 7.8 | 3.6 | 2.9 |

| | | | | |
|---|---|---|---|---|
| João | 4.9 | 7.2 | 4.3 | 3.1 |

Formule o problema como um problema de optimização e resolva-o utilizando CLP.

**PLOP 6: Empresa de Investimentos**

A Administração de uma empresa de investimentos está a considerar 7 diferentes oportunidades de investimento, devendo cada investimento ser feito apenas uma vez. Cada um dos investimentos tem quer um custo quer um ganho a longo prazo próprio, apresentados na tabela abaixo. Os investimentos 1 e 2 são mutuamente exclusivos, tal com os 3 e 4. Para além do mais, nem o investimento 3 nem o 4 podem ser feitos se um dos investimentos 1 ou 2 o não for. Sendo de 100 M€ o montante total de capital disponível para investimento e pretendendo-se maximizar o lucro a longo prazo

| | Oportunidades de Investimento | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Lucro Estimado (M€) | 17 | 10 | 15 | 19 | 7 | 13 | 9 |
| Capital Necessário (M€) | 43 | 28 | 34 | 48 | 17 | 32 | 23 |

Resolva o problema utilizando as capacidades de optimização do SICStus Prolog.

**PLOP 7: Empresa de Transportes**

Uma empresa de transportes tem para despachar num determinado dia 9 encomendas em três camiões. Um programa seleccionou 10 possíveis percursos que os 3 camiões podem seguir, apresentados na tabela abaixo, em que os números 1, 2 e 3 representam a ordem pela qual são entregues as encomendas. A tabela apresenta ainda o tempo total de cada percurso. O objectivo é escolher 3 percursos (um para cada camião) que permitam entregar todas as encomendas num tempo mínimo.

| | Percursos Possíveis | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Cliente | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A | 1 | | | | 1 | | | | 1 | |
| B | | 2 | | 1 | | 2 | | | 2 | 2 |
| C | | | 3 | 3 | | | 3 | | 3 | |
| D | 2 | | | | | 1 | | 1 | | |
| E | | | 2 | 2 | | 3 | | | | |
| F | | 1 | | | 2 | | | 3 | | |
| G | 3 | | | | | | 1 | 2 | | |
| H | | | 1 | | 3 | | | | | 3 |
| I | | 3 | | | | | 2 | | | 1 |
| Tempo | 6 | 4 | 7 | 5 | 4 | 6 | 5 | 3 | 7 | 6 |

**PLOP 8: Caixeiro Viajante**

Considere o problema do caixeiro-viajante. Construa alguns problemas simples de caixeiro-viajante e resolva-os utilizando o SICStus Prolog

**PLOP 9: Fábrica de Brinquedos**

Uma fábrica de brinquedos está a planear a produção de 2 brinquedos para a sua nova campanha de Natal de 2004. Para o brinquedo 1, o custo de início de produção é de 50 k€ gerando um lucro por

brinquedo de 10 €. Para o brinquedo 2 os valores são respectivamente 80 k€ e 15 €. Embora o fabricante tenha duas fábricas foi decidido usar apenas uma delas para evitar duplicar os custos de início de produção. O brinquedo 1 pode ser produzido a um ritmo de 50 por hora na fábrica A e de 40 por hora na fábrica B, enquanto que para o brinquedo 2 os valores respectivos são de 40 e 25. A fábrica A tem 500 horas de produção disponível enquanto que a fábrica B tem 700 horas. Pretende-se encontrar a solução que maximiza o lucro obtido nesta campanha.

**PLOP 10: Sequenciamento de Tarefas**

É necessário fazer uma sequência de 5 tarefas numa máquina, mas o tempo de inicialização (*setup time*) de cada tarefa depende da tarefa anterior, de acordo com a tabela abaixo. Nestas condições, qual a sequência de tarefas que minimiza o conjunto dos tempos de inicialização das tarefas?

| | | Tempo de Inicialização | | | | |
|---|---|---|---|---|---|---|
| | | Tarefa | | | | |
| | | 1 | 2 | 3 | 4 | 5 |
| Tarefa Anterior | Nenhum | 4 | 5 | 8 | 9 | 4 |
| | 1 | | 7 | 12 | 10 | 9 |
| | 2 | 6 | | 10 | 14 | 11 |
| | 3 | 10 | 11 | | 12 | 10 |
| | 4 | 7 | 8 | 15 | | 7 |
| | 5 | 12 | 9 | 8 | 16 | |

**PLOP 11: Produção em Máquinas**

Numa fábrica com 2 máquinas (*machine shop*) constroem-se dois tipos de produtos. Cada unidade do produto A requer 3 horas da máquina 1 e 2 horas da máquina 2, enquanto que uma unidade do produto B requer 2 horas da máquina 1 e 3 horas da máquina 2. A máquina A está disponível 8 horas por dia e a máquina B apenas 7 horas. O lucro obtido na venda de cada unidade dos produtos é de 16 € para o A e 10 € para o B. Tendo de ser um múltiplo inteiro de 0.25 a quantidade produzida de cada produto por dia, pretende-se determinar a quantidade dos dois produtos que maximizam o lucro.

**PLOP 12: Eliminação de Simetrias e Restrições Redundantes**

a) Análise a eficiência dos programas construídos e a sensibilidade desta à variação dos dados de entrada

b) Codifique restrições adicionais em todos os problemas de forma a eliminar simetrias

c) Analise a variação de eficiência na resolução dos problemas anteriores, com e sem a remoção das simetrias.

d) Introduza restrições redundantes nos problemas que permitam aumentar a eficiência de resolução

## Bibliografia

K. Marriot e P. Stuckey, Programming with Constraints – An Introduction, MIT Press, 1998,

E. Tsang, Foundations of Constraint Satisfaction, Academic Press, 1993

Pierre Flener, Constraint Technology for Solving Combinatorial Problems, 2003, [On-Line], Disponível em: http://user.it.uu.se/~pierref/courses/CT/ [consultado em 11/11/2004]

Pedro Barahona, Programação por Restrições, Ano Lectivo 2003/04, Problemas Propostos, 2003 [On-Line], Disponível em: http://ssdi.di.fct.unl.pt/%7Epb/cadeiras/pr/0304/problemas.htm [consultado em 11/11/2004]

SICStus Prolog HomePage, Swedish Institute of Computer Science, [On-Line], Disponível em: http://www.sics.se/isl/sicstus/ [consultado em 15/10/2004]

SICStus Prolog User´s Manual, Swedish Institute of Computer Science, 2003

Learning Constraint Logic programming with Logical Puzzles, [On-Line], Disponível em: http://brownbuffalo.sourceforge.net/, [consultado em 11/11/2004]